

TYPO LOGO

TYPO LOGO

Dun Dun.

타이포 로고 시안

1. 공연장의 스피커가 울릴 때의 소리를 '둔둔'이라는 의성어로 표현
2. 노래의 울림에는 강약이 있으며, 이를 타이포의 굵기를 조절하여 나타냄
3. 아티스트의 공지에 사용될 이미지와, 아티스트 메인 이미지 등의 컬러를 고려하여 흑백
으로 제작

추가로 아이콘 로고를 디자인하여 앱 실행 시, 혹은 메인 이미지로 사용할지의 여부 결정할 예정

TYPO FACE

Futura PT Light

Aa Bb Cc Dd Ee Ff Gg Hh Ii Jj Kk
Ll Mm Nn Oo Pp Qq Rr Ss Tt Uu Vv
Ww Xx Yy Zz
123456789!@/\$%^&*()

Futura PT Book

Aa Bb Cc Dd Ee Ff Gg Hh Ii Jj Kk
Ll Mm Nn Oo Pp Qq Rr Ss Tt Uu Vv
Ww Xx Yy Zz
123456789!@/\$%^&*()

TYPO FACE

Pretandard Variable Black

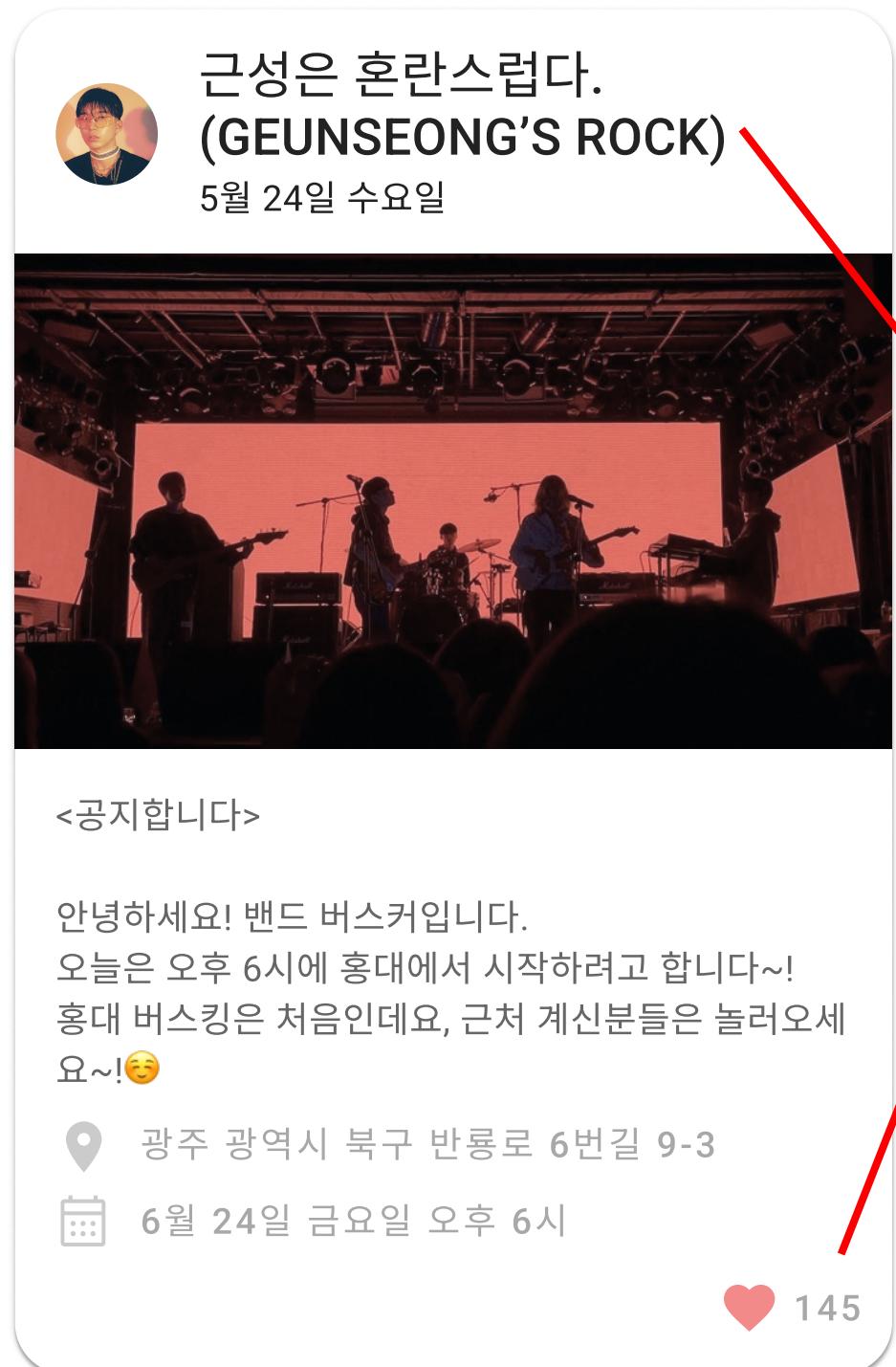
Aa Bb Cc Dd Ee Ff Gg Hh Ii Jj Kk
Ll Mm Nn Oo Pp Qq Rr Ss Tt Uu Vv
Ww Xx Yy Zz
123456789!@/\$%^&*()

Pretandard Variable Semibold

Aa Bb Cc Dd Ee Ff Gg Hh Ii Jj Kk
Ll Mm Nn Oo Pp Qq Rr Ss Tt Uu Vv
Ww Xx Yy Zz
123456789!@/\$%^&*()

STORYBOARD

Dun Dun.



USER FLOW

- 로그인 이후의 시작 지점
- 아티스트 이미지 로고를 누르면, 아티스트 페이지로 넘어감
- 하단의 탭으로 검색, 캘린더, 맵, 환경 설정으로 이동 가능

Description.

1. 하단 바

- 좋아요
하트를 누르면 회색에서 빨간색으로 색상변경
하트 옆에 개수 표시 되어야 함

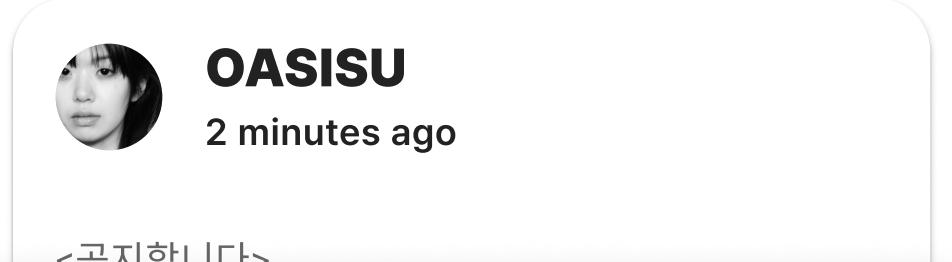
2. 상단 바

- 아티스트 이미지 로고 (→ 아티스트 페이지로 이동)
- 아티스트 이름
- 작성 날짜

3. 공지 내용

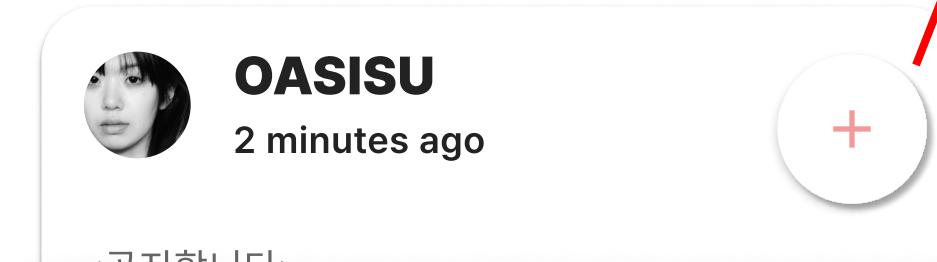
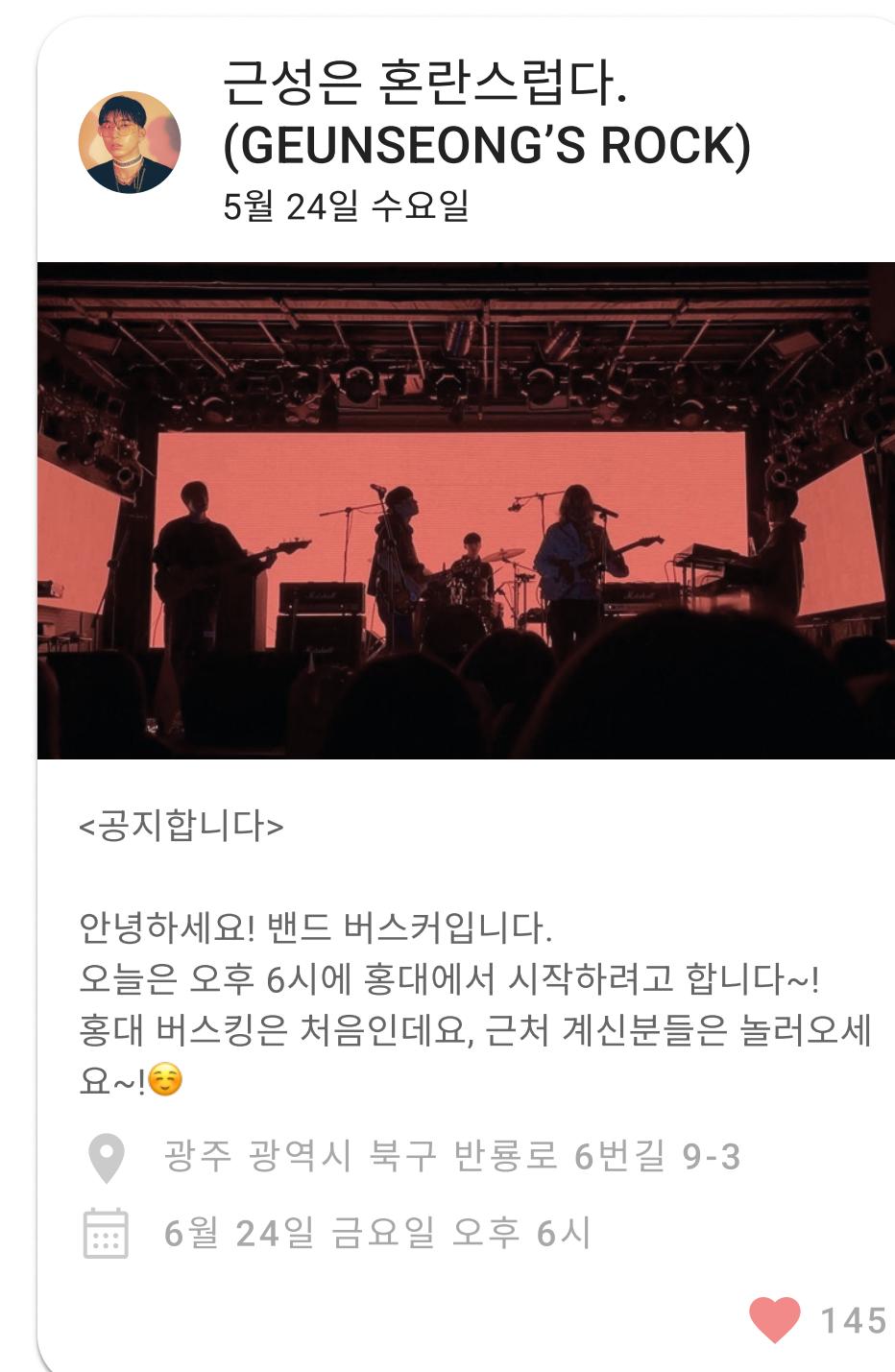
- 사진 표시(입력 받았을 때에만)
- 공지 글 표시(필수)
- 지도 표시(입력 받았을 때에만)
- 캘린더 표시(입력 받았을 때에만)

• 더보기



Home Search Calendar Maps Settings

Dun Dun.



Home Search Calendar Maps Settings

USER FLOW

- 로그인 이후의 시작 지점
- 아티스트 이미지 로고를 누르면, 아티스트 페이지로 넘어감
- 하단의 탭으로 검색, 캘린더, 맵, 환경 설정으로 이동 가능
- 우측 하단의 버튼으로
내 아티스트 페이지로 이동
공지 작성 페이지로 이동

Description.

1. 우측 하단 버튼

- 첫번째 터치 이후, 마이페이지와 Writting 버튼이 생성
- 마이페이지 → 아티스트 자신의 아티스트 페이지로 이동
- Writting → 아티스트 공지 작성 페이지로 이동

2. 그외

- 그 외로는 모두 유저의 페이지와 같다.

My Page
Writting

STORYBOARD



USER FLOW

1. HOME에서 아티스트 이미지 로고를 누르면 넘어옴.
2. 유저는 본 화면에서 아티스트가 이전에 작성한 공지들을 스크롤을 하여 확인할 수 있음.
3. 이전 버튼으로 다시 HOME 화면

Description.

1. Follow

- 언팔 상태에서는 회색 → 팔로우 상태에서는 색상
- FOLLOW 버튼을 누르는 것으로 상태가 변경됨

2. 공지내용

- 아티스트가 작성한 모든 공지글을 스크롤을 통해서 확인할 수 있음
- 이 순서는 최신 작성 순이 맨 위에 있어야 함.



USER FLOW

<자신의 아티스트 페이지>

1. 우측 하단 버튼으로 마이페이지 이동하는 경우
2. 아티스트가 HOME에서 자신의 프로필 사진을 누른 경우

Description.

1. 프로필 화면

- Edit 누를 경우→
- 소개글 변경 + 사진 변경

2. 공지내용

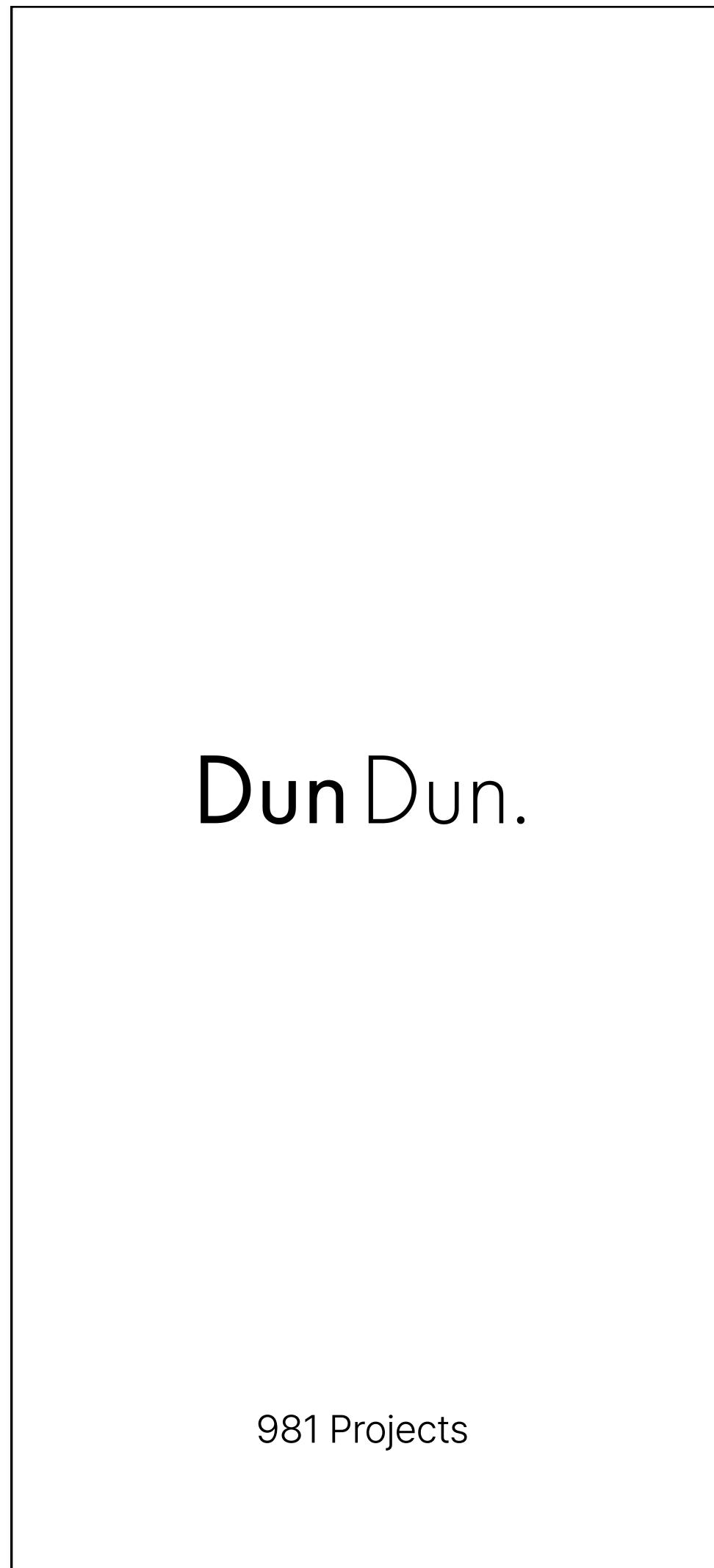
- 공지내용의 우측 상단의 설정 버튼을 통해, 아티스트 공지 작성 페이지로 이동

3. 상단제목

- 아티스트 이름 (애니메이션)
- 하단의 "here is my great~"은 고정글귀



STORYBOARD



Dun Dun.

981 Projects

A screenshot of a mobile application's sign-up screen. The title "Dun Dun." is at the top. Below it is a "아이디" (ID) field with placeholder text "아이디를 입력해주세요!" and a green checkmark indicating it's a valid ID. A note below says "사용 가능한 아이디예요:". There are three other fields: "닉네임" (Nickname) with placeholder "8자 이내 한글 혹은 영문" and a red exclamation mark warning "사용할 수 없는 닉네임이에요:", "비밀번호" (Password) with placeholder "8~20자리 영대-소문자, 숫자 조합" and a green checkmark indicating it's a valid password; and "비밀번호 확인" (Password Confirmation) with placeholder "8~20자리 영대-소문자, 숫자 조합" and a green checkmark indicating it matches the password above.

Dun Dun.

아이디

아이디를 입력해주세요!

✓ 사용 가능한 아이디예요:

닉네임

8자 이내 한글 혹은 영문

! 사용할 수 없는 닉네임이에요:

비밀번호

8~20자리 영대-소문자, 숫자 조합

✓ 사용 가능한 비밀번호예요:

비밀번호 확인

8~20자리 영대-소문자, 숫자 조합

✓ 비밀번호가 일치합니다:

A screenshot of the same mobile application's sign-up screen, showing the second step where the user enters their artist name. The title "Dun Dun." is at the top. Below it is a "아티스트 이름" (Artist Name) field with placeholder text "8자 이내 한글 혹은 영문" and a green checkmark indicating it's a valid artist name. A note below says "사용 가능한 아티스트 이름이에요:". There are two other fields: "비밀번호" (Password) with placeholder "8~20자리 영대-소문자, 숫자 조합" and a green checkmark indicating it's a valid password; and "비밀번호 확인" (Password Confirmation) with placeholder "8~20자리 영대-소문자, 숫자 조합" and a green checkmark indicating it matches the password above.

Dun Dun.

아이디

아이디를 입력해주세요!

✓ 사용 가능한 아이디예요:

아티스트 이름

8자 이내 한글 혹은 영문

! 사용할 수 없는 아티스트 이름이에요:

비밀번호

8~20자리 영대-소문자, 숫자 조합

✓ 사용 가능한 비밀번호예요:

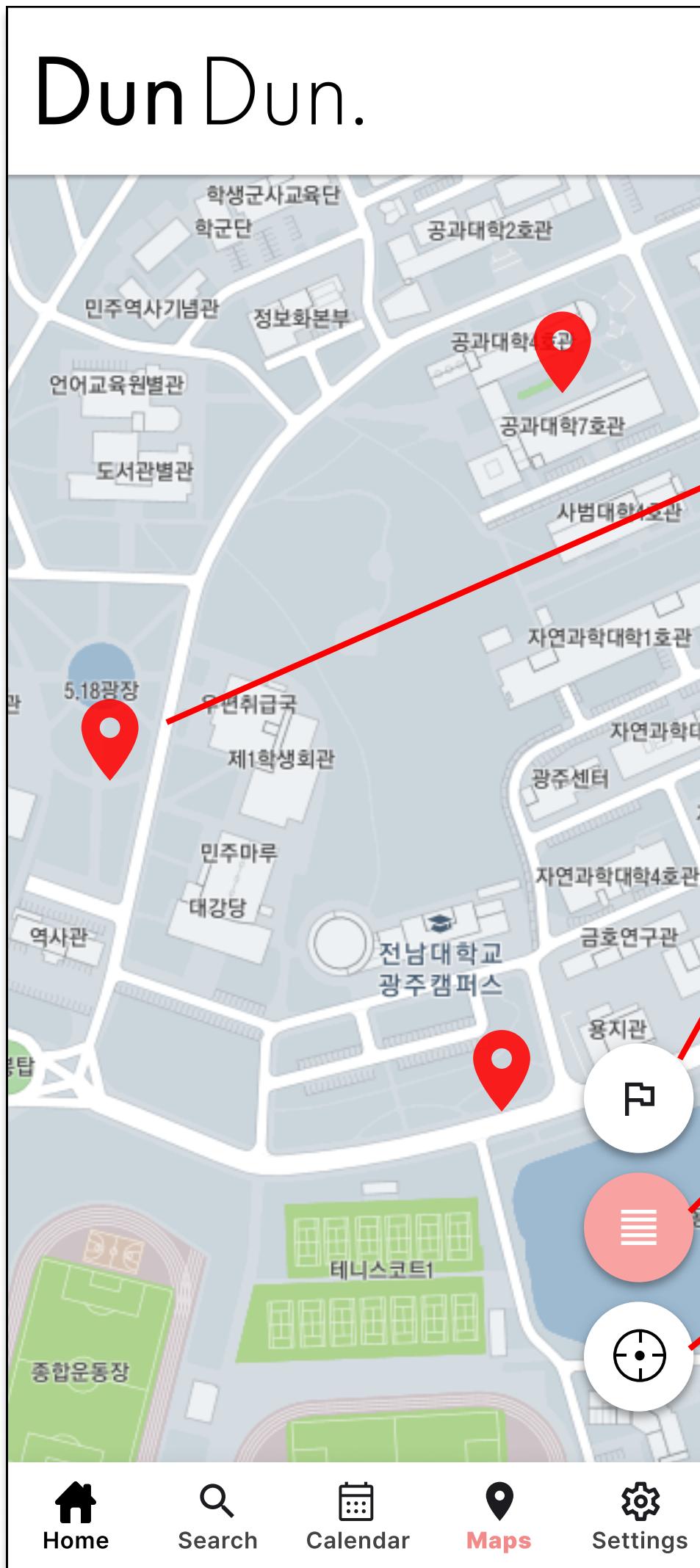
비밀번호 확인

8~20자리 영대-소문자, 숫자 조합

✓ 비밀번호가 일치합니다:

981Projects

STORYBOARD



USER FLOW

- 하단 탭에서 지도 버튼 눌렀을 경우에 본 페이지로 이동.

Description.

1. 기본 세팅

- 자신의 위치에 기반하여 가까운 공연 장소를 빨간색 마커로 보여준다.
- 지도를 움직이면 그와 관련된 공연 장소를 클러스팅 해서 보여준다.

2. Follow

- 기본 세팅과 똑같지만 버튼을 누르면, 팔로우한 아티스트만 지도에서 표기한다.
- 버튼을 눌렀을 때 색상 변경.



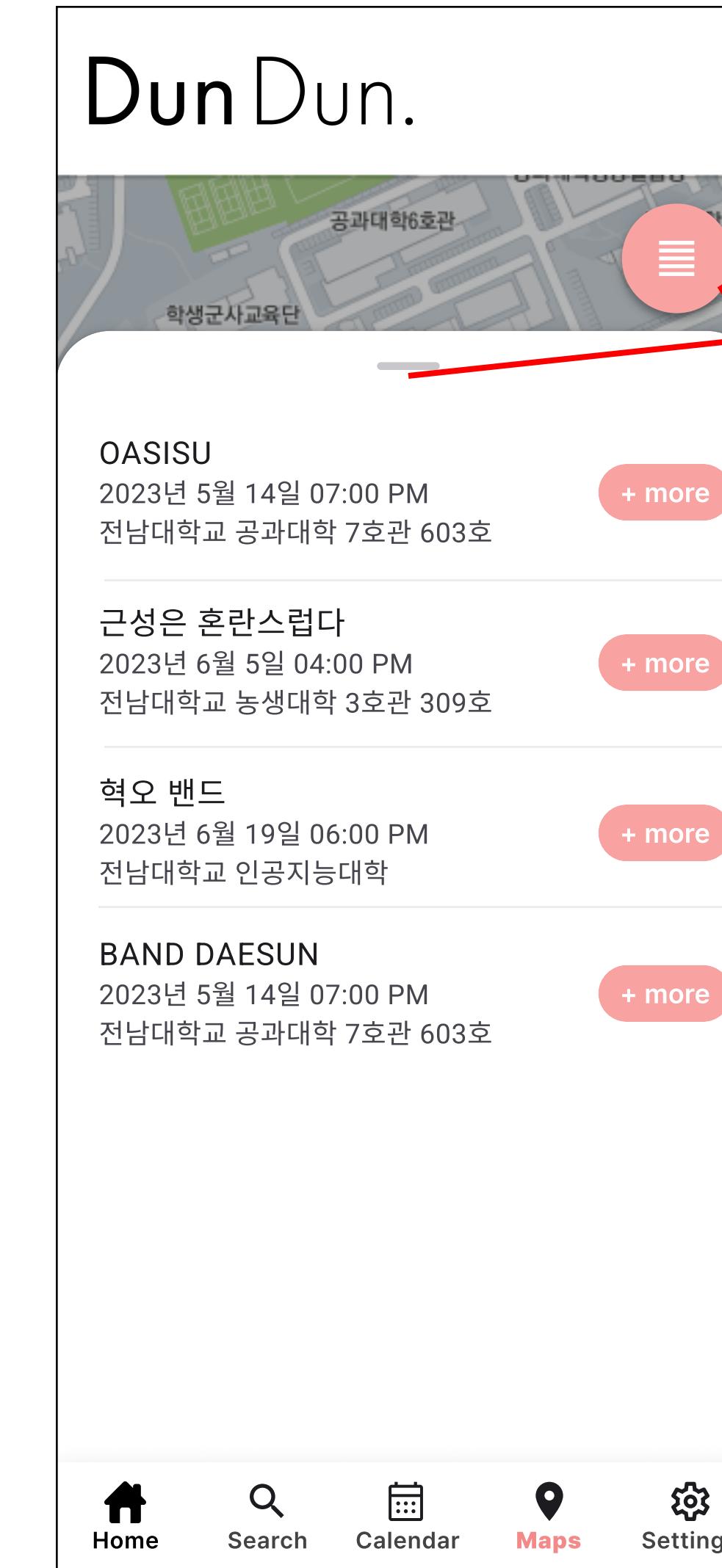
3. List

- 현재 지도를 중심으로 하여서, 표기되는 마커들의 상세 정보를 보여준다. (옆의 화면)

4. 현재 위치

- 지도를 멀리 이동하였을 때, 현재 위치로 돌아옴.

Red arrows from the 'List' and 'Current Location' sections point to the 'List' section of the second storyboard screen.



USER FLOW

- 리스트 내에서의 +more 버튼을 눌렀을 경우 아티스트 페이지로 이동
- (1) 상세 정보를 밑으로 쓸거나,
(2) 리스트버튼을 다시 누르면,
→ #7 지도로 변경

Description.

1. List

- #7 지도 페이지의 리스트 버튼을 누르면 색상이 #8처럼 변경된다.



2. +more

- 버튼 누르면 아티스트 페이지로 이동.

DEVELOP

	Author ▾	Label ▾	Projects ▾	Milestones ▾	Assignee ▾	Sort ▾
3 Open ✓ 6 Closed						
[BE] mock 데이터 생성 FE feature						
#11 opened last week by rootachieve 1 task						
[BE] UseCase 작성 BE feature						
#10 opened last week by rootachieve 2 tasks						
[FE] 공지 페이지 리사이클러뷰 작업 FE feature						
#9 opened last week by ajang369 3 tasks						

	Author ▾	Label ▾	Projects ▾	Milestones ▾	Assignee ▾	Sort ▾
3 Open ✓ 6 Closed						
[FE] 패키지 변경 및 뷰바인딩 설정 FE feature refactor				↑ 1		
#14 by rootachieve was closed last week 3 tasks done						
[BE] DTO 생성 BE feature				↑ 1		
#8 by rootachieve was closed 2 days ago 3 tasks done						
[FE] 지도 페이지에 카카오맵 연동 done FE				↑ 1		
#7 by ajang369 was closed 2 days ago 3 tasks done						
[FE] 하단 네비게이션 바 배치 done FE				↑ 1		
#6 by ajang369 was closed last week 3 tasks done						
[BE] Firebase 사용 설정 BE feature init				↑ 1		
#5 by rootachieve was closed last week 3 tasks done						
[Init] 프로젝트 생성 done feature init				↑ 1		
#1 by rootachieve was closed on May 22						

COMMON

DEVELOP

■ 표

DUNDUN

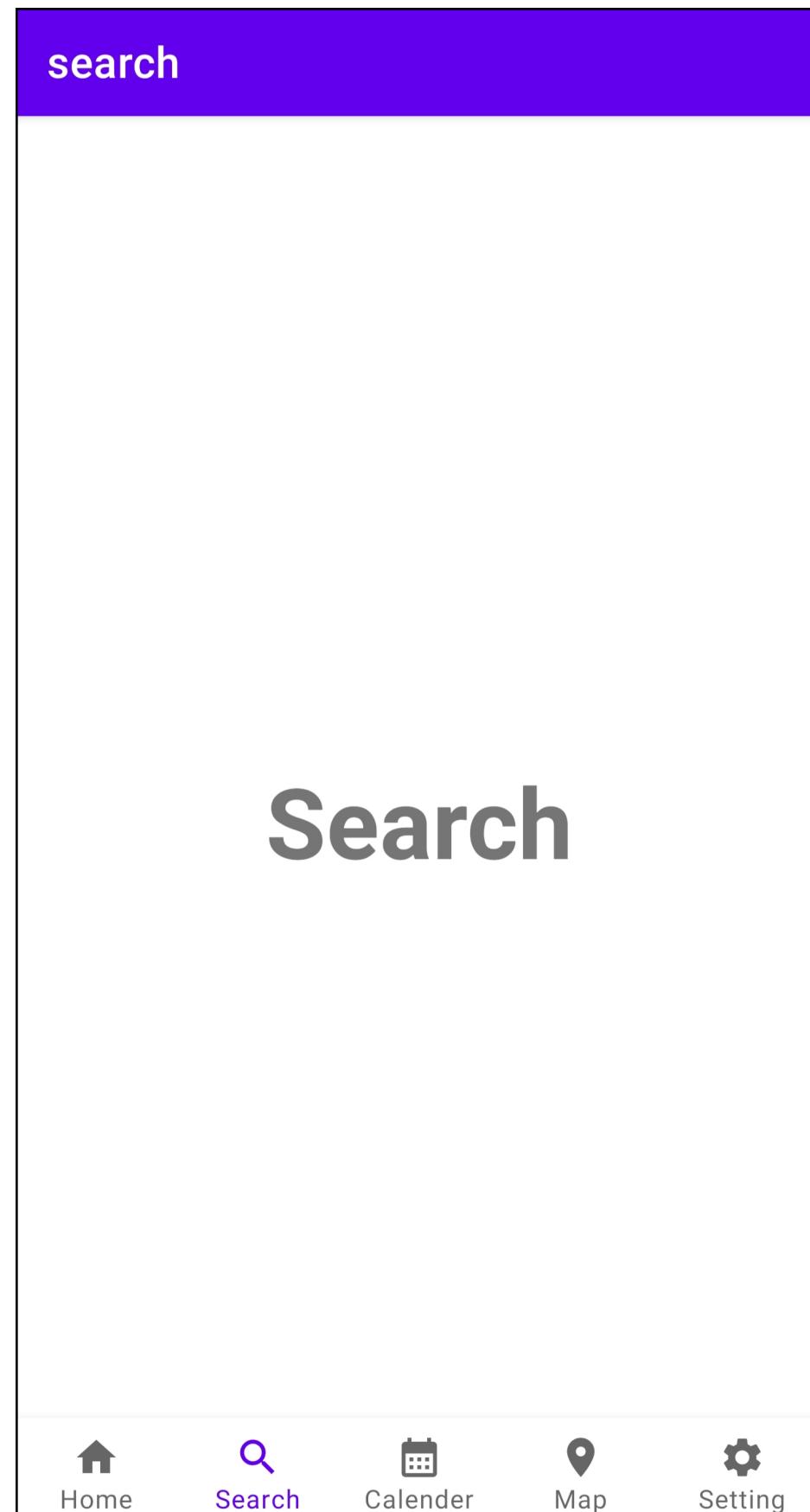
대분류	중분류	Aa 소분류	부가사항	우선순위	상태	...
회원가입	스플래쉬	스플래쉬	둔둔 로고 이미지	2	Not started	
	로그인	아이디 / 비밀번호 입력	- 로그인 상태 유지 기능		Not started	
		하단 바	- 회원 가입 - 아이디 / 비밀번호 찾기		Not started	
회원가입	유저 / 아티스트 구분				Not started	
	회원 가입 입력창		- 아이디 - 닉네임 / 아티스트 이름 (아티스트가 자기 자신을 팔로우하게) (근성) - 비밀번호 - 비밀번호 확인		Not started	
온보딩				4	Not started	
온보딩	온보딩 애니메이션				Not started	
아티스트 추천	아티스트 페이지 이동	거리 순 기반의 아티스트 추천			Not started	
	다음으로 넘어가기 기능				Not started	
HOME				1	Not started	
	게시글	좋아요 누르기	좋아요 개수 표시		Not started	
		시간	사용자 기준 몇분전 표시		Not started	
		캘린더	(없으면 표시 안함)		Not started	
		지도	(없으면 표시 안함)		Not started	
		아티스트	아티스트 페이지 누르면 아티스트 페이지로.		Not started	
아티스트 페이지	아티스트 사진				Not started	
		아티스트 이름 - 소개글			Not started	
		Follow 기능	아티스트는 Follow가 아니라 EDIT		Not started	
		EDIT 기능	EDIT는 아티스트 사진과 이름- 소개글 수정 가능		Not started	
		아티스트 공지 글	하단에 슬라이드 형식으로 아티스트가 작성한 글		Not started	
아티스트용 토클	공지사항 작성		우측하단 위치		Not started	
아티스트용 공지작성	텍스트 작성		제목 기능 있어야함		Not started	
		캘린더 입력 기능			Not started	
		지도 입력 기능	검색 창을 통해서 지도로 연결 되게끔		Not started	

Design

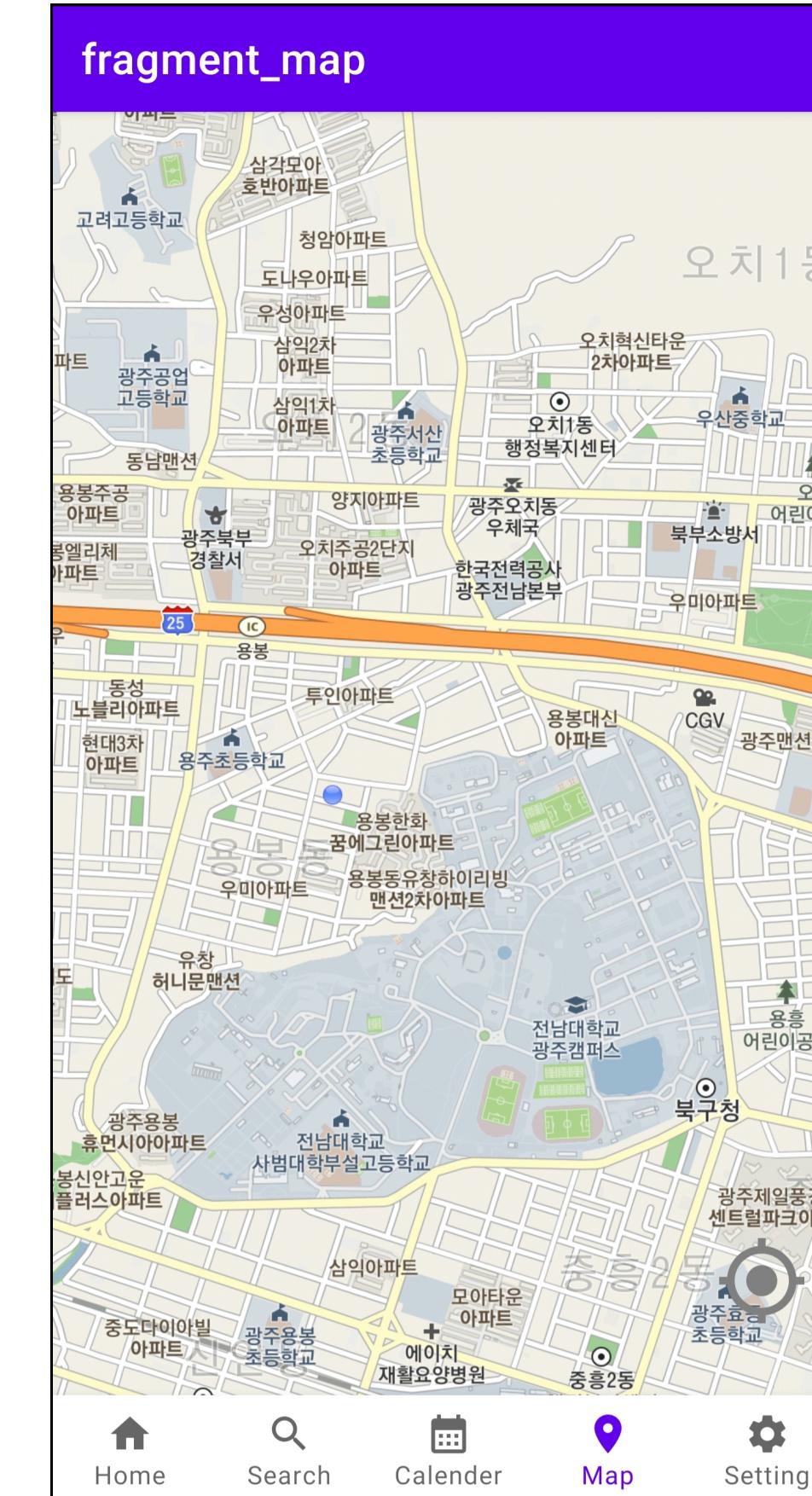
Storyboard

Develop

DEVELOP



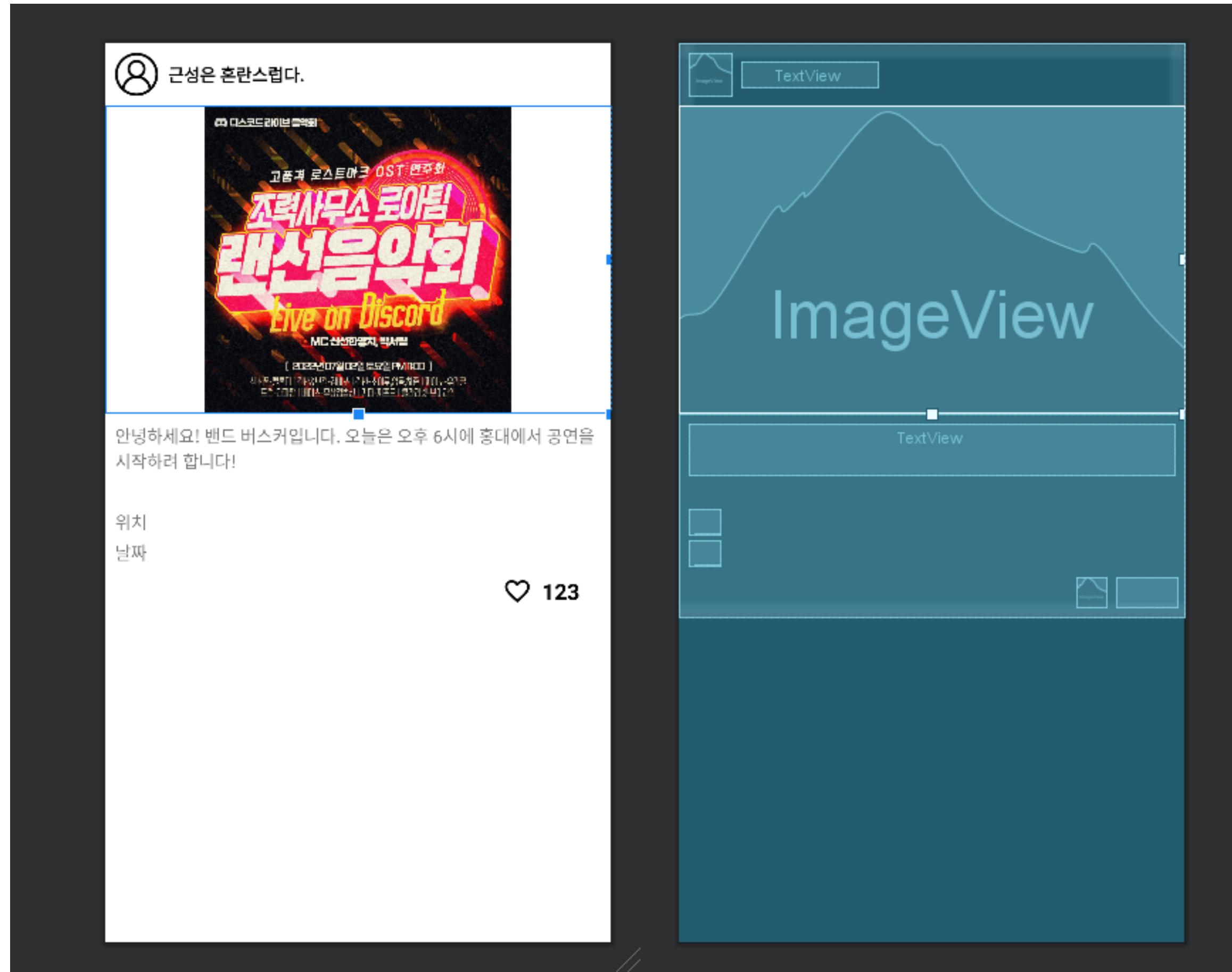
하단 네비게이션 바
프레그먼트 사용



카카오맵 api 적용

GPS 받아서 내 위치로 이동

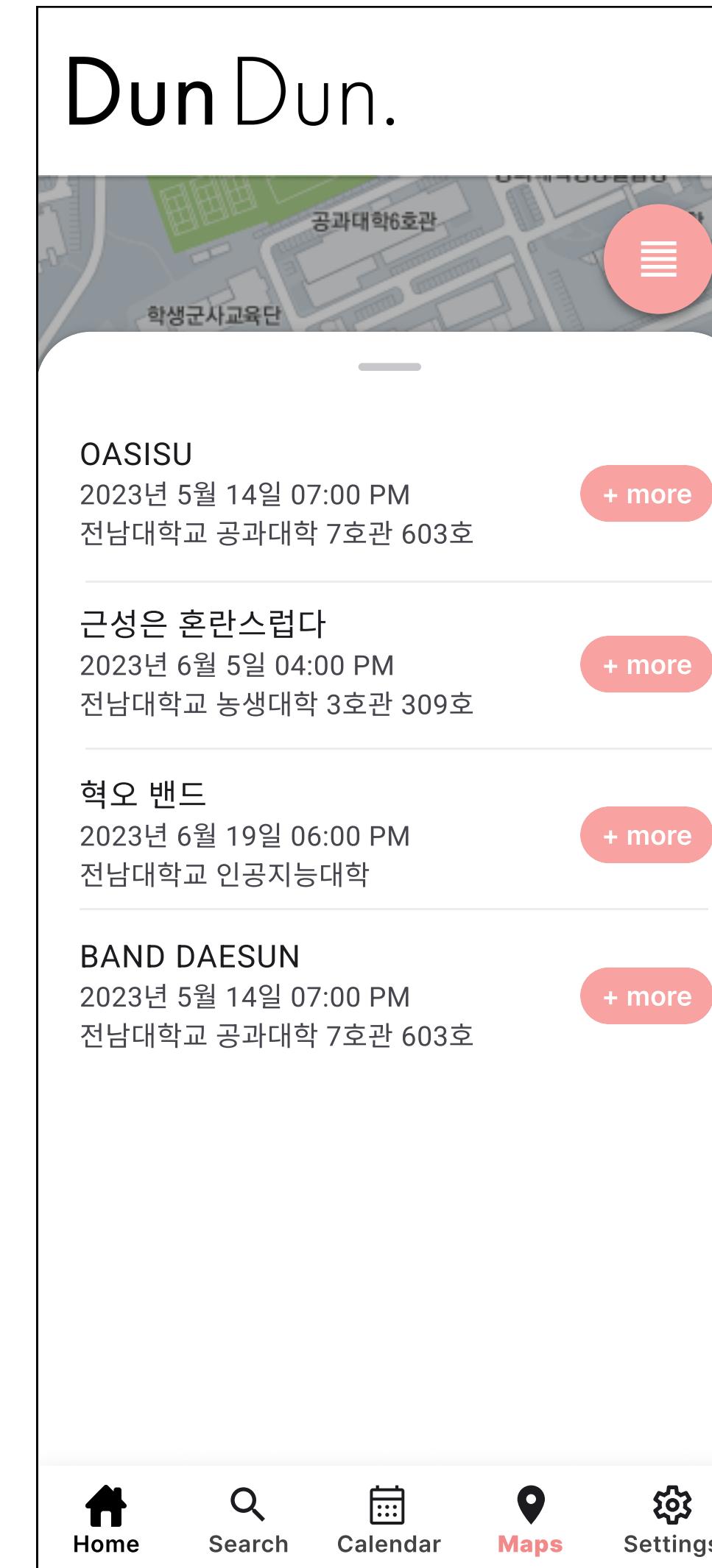
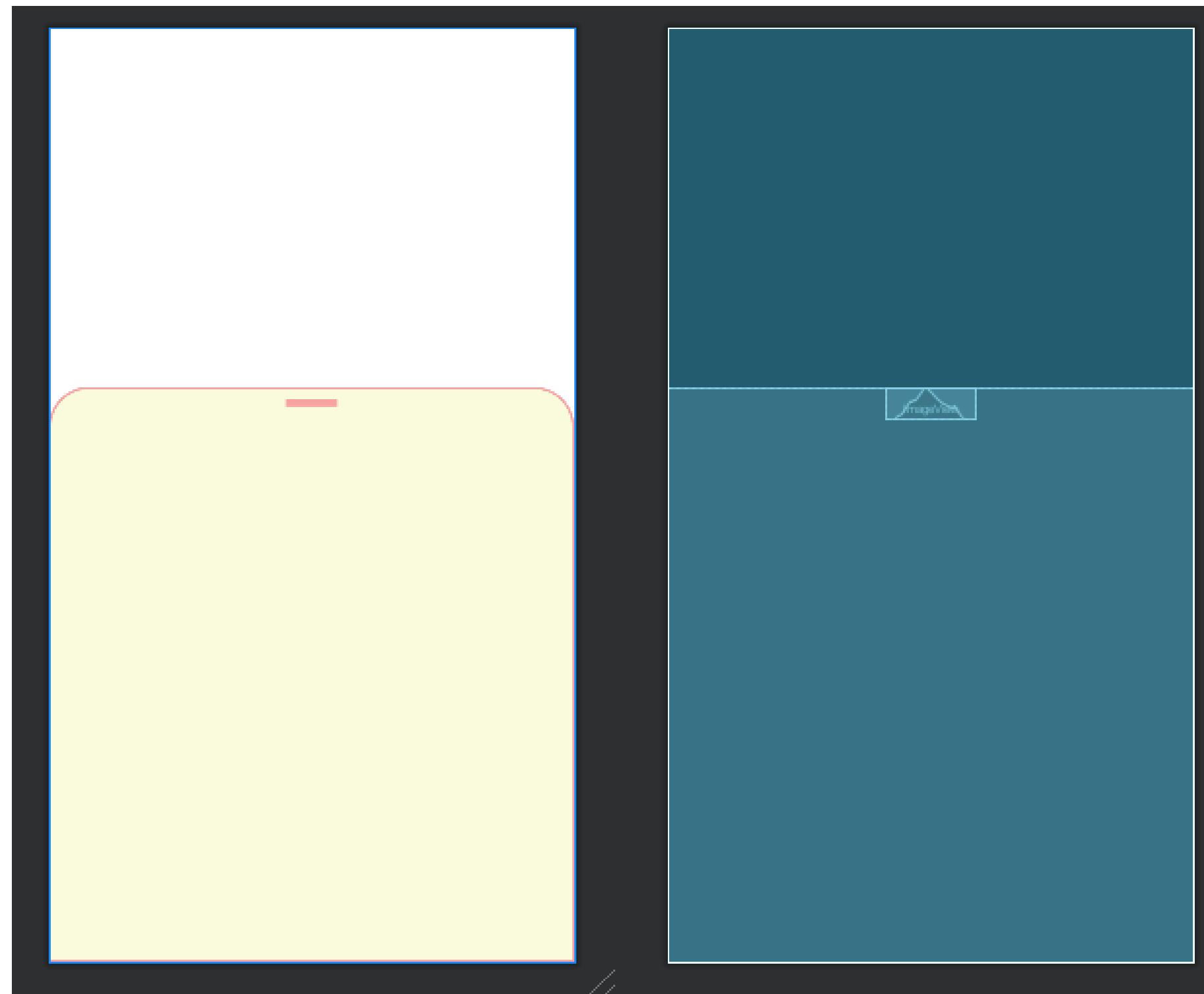
DEVELOP



HOME 화면 게시글 레이아웃

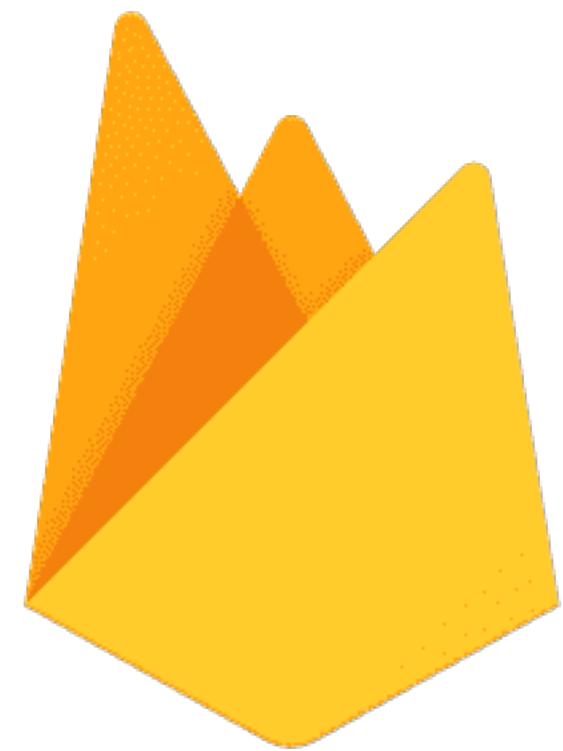
게시글 스크롤 기능 작업 중

DEVELOP



Map 화면에서 공연 목록

Bottom sheet 작업 중

DEVELOP

Firebase

DEVELOP

구글에서 제공하는
모바일, 웹 애플리케이션 개발 플랫폼

DEVELOP

구글에서 제공하는
모바일, 웹 애플리케이션 개발 플랫폼

Auth
DB
Storage
Analytics
A / B Testing

....

DB

Design

Storyboard

Develop

DEVELOP

DEVELOP

Firestore은 noSQL이다

```
data class NoticeDTO(  
    val artistId: String,  
    val articleId: String,  
    val noticeTitle: String,  
    val noticeContent: String,  
    val noticeImage: String,  
    val likeCount: Long,  
    val date: Timestamp,  
    val geo: GeoPoint,  
    val geoHash: String,  
    val locationDescription: String,  
    val createTime: Timestamp,  
    val updateTime: Timestamp,  
)
```

```
data class NoticeDisplayDTO(  
    val articleId: String,  
    val artistName: String,  
    val profileImageUrl: String,  
    val createDate: Date,  
    val contentImageURL: String,  
    val noticeTitle: String,  
    val content: String,  
    val locationDescription: String,  
    val date: Date,  
    val likeCount: Int,  
)
```

DEVELOP

Firestore은 noSQL이다

```
Firebase.firebaseio FirebaseFirestore
    .collection( collectionPath: "Email" ) CollectionReference
        .document( email ) DocumentReference
            .get().addOnCompleteListener { it: Task<DocumentSnapshot!>
                if (it.isSuccessful) {
                    if (it.result.exists()) {
                        callback(Result.success( value: true))
                    } else {
                        callback(Result.success( value: false))
                    }
                } else {
                    callback(Result.failure(Exception("서버와 연결에 실패했습니다.")))
                }
            }
        }
```

DEVELOP

auth가 매우 간편하다!

```
auth.createUserWithEmailAndPassword(email, pw).addOnCompleteListener { task ->
    if (task.isSuccessful) {
```

```
[auth.currentUser?.uid,
```

DEVELOP

```
class UserSignUpUseCase {
    operator fun invoke(
        email: String,
        pw: String,
        name : String,
        callback: (Result<Boolean>) -> (Unit)
    ) {
        return MainRepository.createUser(email, pw, name, callback)
    }
}
```

DEVELOP

**Value animator와
Object animator를 통해
뷰에 애니메이션 집어넣기**

DEVELOP

Value Animator

비동기적으로 시작 값에서 종료 값으로
값을 점차 변화

변하는 값을 콜백을 통해 알려줌

DEVELOP

Async 0 → 255



sync

시작값, 종료값
콜백 전달

DEVELOP

Async 0 → 255

sync
시작값, 종료값
콜백 전달

콜백 호출

DEVELOP

Async 0 → 255

sync
시작값, 종료값
콜백 전달

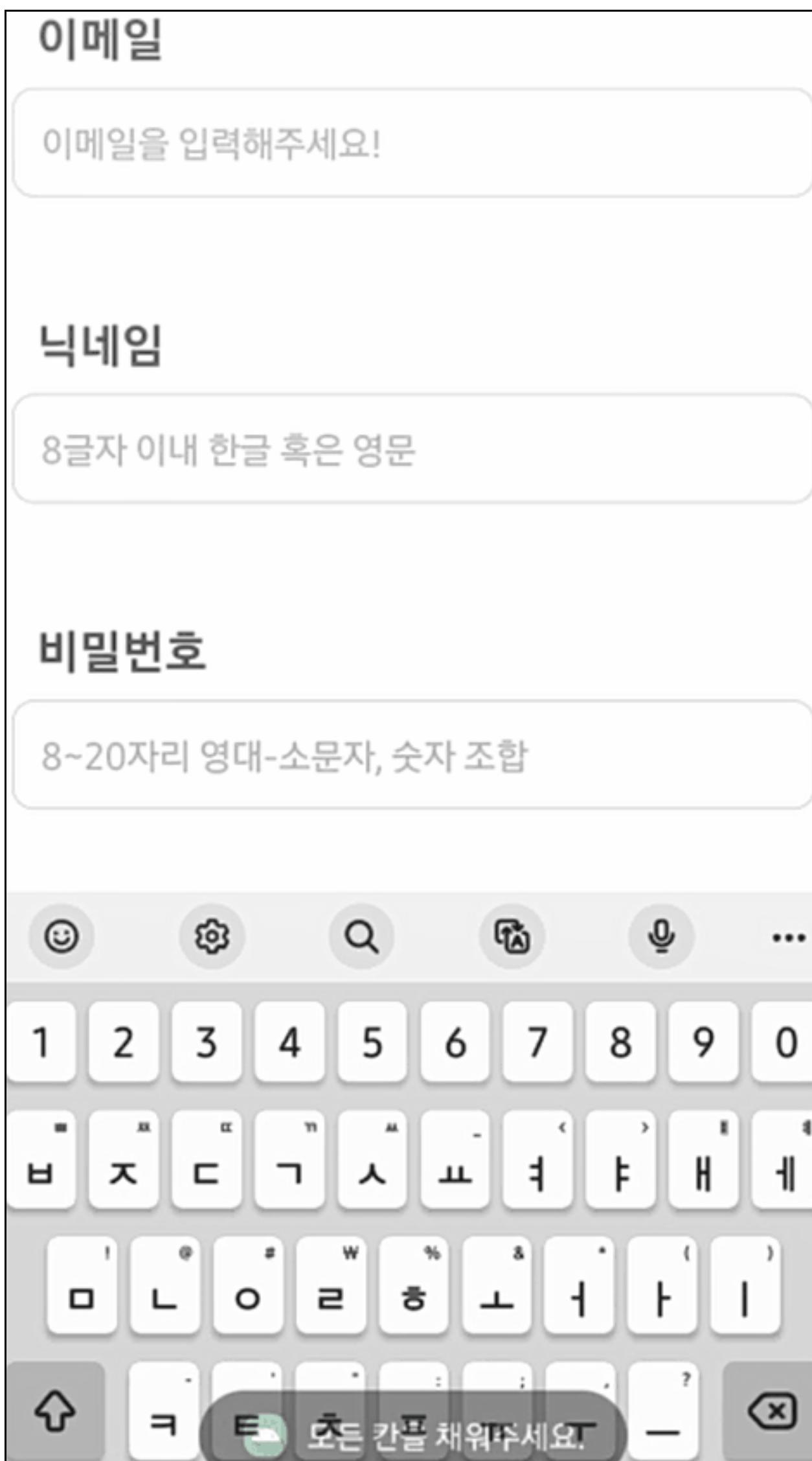
콜백 호출

```
startAnimator = ValueAnimator.ofFloat(...values: 0f, 40f).apply { this: ValueAnimator!
    addUpdateListener { it: ValueAnimator
        size = it.animatedValue as Float
        textPaint.color = Color.WHITE
    }
    duration = 100L
    repeatCount = 0
    interpolator = AccelerateDecelerateInterpolator()
}
```

```
startAnimator = ValueAnimator.ofFloat(...values: 0f, 40f).apply { this: ValueAnimator!
    addUpdateListener { it.콜백 animator
        size = it.animatedValue as Float
        textPaint.color = Color.WHITE
    }
    duration = 100L      지속시간
    repeatCount = 0      반복횟수
    interpolator = AccelerateDecelerateInterpolator() 보간
}
```

범위

DEVELOP



DEVELOP

Object Animator

비동기적으로 시작 값에서 종료 값으로
값을 점차 변화

변하는 값을 오브젝트에 적용시킴

DEVELOP

Value Animator와 Object Animator의 차이

오브젝트에 바로 적용시키느냐
or
값을 반환해 주느냐

DEVELOP

```
val animator: ObjectAnimator = ObjectAnimator.ofFloat( target: this, propertyName: "rotationY", ...values: 0f, 360f)
animator.duration = 700
animator.start()
```

DEVELOP

							8
1	2	3	4	5	6	7	
.	8	9	10	11	12	13	14
15	16	17	18	19	20	21	
22	23	24	25	26	27	28	
29	30	31					

기술조사

Design

Storyboard

Develop

DEVELOP

